Proof Complexity of Practical Integer Programming

Noah Fleming

Based on

« On the power and Limitations of Branch and Cut 
  Fleming, Göös, Impagliazzo, Pitassi, Robere, Wigderson »
Outline

★ Using Proof Complexity for Algorithm Analysis
★ Proof Complexity of Integer Programming
  - Cutting Planes
★ Branch and Cut and Stabbing Planes
★ Stabbing Planes vs. Cutting Planes
Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms $A$ as a proof system $P$. 
Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms $A$ as a proof system $P$.

$\triangleright$ Hides practical details of algorithms
Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms $A$ as a proof system $P$

▷ Hides practical details of algorithms

▷ Lower bounds on $P$-proofs $\Rightarrow$ lower bounds on runtime of $A$
Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms $A$ as a proof system $P$

- e.g. Algorithms for SAT
  - CDCL and Resolution
Algorithm Analysis from Proofs

Idea: Formalize the techniques used in a class of algorithms $A$ as a proof system $P$

- e.g. Algorithms for SAT
  - CDCL and Resolution
- e.g. Algorithms for Integer Programming
  - Chvátal-Gomory Cutting Planes and Cutting Planes
Integer Programming

Integer-programming: Given $Ax \geq b$, find $x \in \mathbb{Z}^n$, $Ax \geq b$
Integer Programming

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$
Integer Programming

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$

A classic approach: Chvátal-Gomory Cutting Planes
Chvátal–Gomory Cutting Planes

Integer-programming: Given $P = \{x: Ax \geq b\}$ find $x \in \mathbb{P} \cap \mathbb{Z}^n$

A classic approach: Chvátal–Gomory Cutting Planes

CG-Cut: If $\alpha x \geq b$ is valid for $P$
Chvátal–Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in P \cap \mathbb{Z}^n$.

A classic approach: Chvátal–Gomory Cutting Planes

$CG$-Cut: If $ax \geq b$ is valid for $P$.

$a \in \mathbb{Z}^n$, $b \in \mathbb{R}$.
Chvátal-Gomory Cutting Planes

Integer-programming: Given $P = \{x : Ax \geq b\}$ find $x \in \mathbb{P} \cap \mathbb{Z}^n$

A classic approach: Chvátal-Gomory Cutting Planes

CG-Cut: If $ax \geq b$ is valid for $P$ then $ax \geq \lceil b \rceil$ is a CG-cut
Chvátal–Gomory Cutting Planes

Integer-programming: Given \( P = \{ x : Ax \geq b \} \), find \( x \in \mathbb{P} \cap \mathbb{Z}^n \)

A classic approach: Chvátal–Gomory Cutting Planes

CG-Cut: If \( ax \geq b \) is valid for \( P \), then \( ax \geq \lceil b \rceil \) is a CG-cut

→ Preserves integer solutions to \( P \)
Chvátal-Gomory Cutting Planes

**Integer-programming:** Given $P = \{ x : A x \geq b \}$ find $x \in P \cap \mathbb{Z}^n$

---

**CG - Cutting Planes**

Heuristically add CG-cuts to $P$

**CG-Cut:** If $a x \geq b$ is valid for $P$

then $a x \geq \lceil b \rceil$ is a CG-cut

$\rightarrow$ Preserves integer solutions to $P$
Chvátal–Gomory Cutting Planes

Integer-programming: Given \( P = \{ x : Ax \geq b \} \) find \( x \in \mathbb{P} \cap \mathbb{Z}^n \)

CG - Cutting Planes
- Heuristically add CG-cuts to \( P \)

CG-Cut: If \( ax \geq b \) is valid for \( P \)
- then \( ax \geq \lceil b \rceil \) is a CG-cut
- Preserves integer solutions to \( P \)
Chvátal–Gomory Cutting Planes

Integer-programming: Given \( P = \{ x : Ax \geq b \} \) find \( x \in \mathbb{P} n \mathbb{Z}^n \)

CG - Cutting Planes

- Heuristically add CG-cuts to \( P \) until:
  - an integer solution is found
  - the empty polytope is deduced
Cutting Planes \[\text{[CCT87]}\]

Let \( P = \{Ax \geq b\} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \)
Cutting Planes

Let \( P = \{ Ax \geq b \} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \)

A CP proof that \( P \cap \mathbb{Z}^n = \emptyset \) is a sequence of polytopes \( P = P_1, \ldots, P_s = \emptyset \)

s.t. \( P_{i+1} \) is deduced from \( P_i \) by a CG-cut
Cutting Planes

Let \( P = \{ A x \geq b \} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \).

A CP proof that \( P \cap \mathbb{Z}^n = \emptyset \) is a sequence of polytopes \( P = P_1, \ldots, P_s = \emptyset \).

s.t. \( P_{i+1} \) is deduced from \( P_i \) by a CG-cut.
Cutting Planes

Let \( P = \{Ax > b\} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \)

A CP proof that \( P \cap \mathbb{Z}^n = \emptyset \) is a sequence
of polytopes \( P = P_1, \ldots, P_s = \emptyset \)

s.t. \( P_{i+1} \) is deduced from \( P_i \) by a CG-cut
Cutting Planes

Let $P = \{Ax > b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

s.t. $P_{i+1}$ is deduced from $P_i$ by a CG-cut
Cutting Planes

Let \( P = \{Ax \geq b\} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \).

A CP proof that \( P \cap \mathbb{Z}^n = \emptyset \) is a sequence of polytopes \( P = P_1, \ldots, P_s = \emptyset \) such that \( P_{i+1} \) is deduced from \( P_i \) by a CG-cut.
Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}_n = \emptyset$

A CP proof that $P \cap \mathbb{Z}_n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

s.t. $P_{i+1}$ is deduced from $P_i$ by a CG-cut.
Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

such that $P_{i+1}$ is deduced from $P_i$ by a CG-cut
Cutting Planes

Let $P = \{Ax > b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$.

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

s.t. $P_{i+1}$ is deduced from $P_i$ by a CG-cut.
Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{Z}^n = \emptyset$.

A CP proof that $P \cap \mathbb{Z}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

s.t. $P_{i+1}$ is deduced from $P_i$ by a CG-cut.
Cutting Planes

Let \( P = \{ Ax \geq b \} \) be such that \( P \cap \mathbb{Z}^n = \emptyset \).

A CP proof that \( P \cap \mathbb{Z}^n = \emptyset \) is a sequence of polytopes \( P = P_1, \ldots, P_s = \emptyset \)

s.t. \( P_{i+1} \) is deduced from \( P_i \) by a CG-cut.
Cutting Planes

Let $P = \{Ax \geq b\}$ be such that $P \cap \mathbb{N}^n = \emptyset$

A PCP proof that $P \cap \mathbb{N}^n = \emptyset$ is a sequence of polytopes $P = P_1, \ldots, P_s = \emptyset$

S.t. $P_{i+1}$ is derived from $P_i$ by a CG-cut

Proof size: the number of polytopes $s$
Cutting Planes

Powerful & Well-studied algebraic proof system
Cutting Planes

Powerful & Well-studied algebraic proof system

Short proofs of PHP
Cutting Planes

Powerful & Well-studied algebraic proof system

▷ Short proofs of P≠P

▷ Exponential lower bounds [Pud97]
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure
Integer Programming

Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

Branch-and-Cut on input $P$,

Branch: break $P$ into $P_1, \ldots, P_k$

s.t. $\mathbb{P} \mathbb{N} \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

**Branch-and-Cut** on input P,

**Branch:** break $P$ into $P_1, \ldots, P_k$

s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

**Branch-and-Cut** on input $P$,

Branch: break $P$ into $P_1, \ldots, P_k$

s.t. $P \cap \mathbb{Z}^n \subseteq P_1 \cup \ldots \cup P_k$
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure:

**Branch-and-Cut**

On input $P$, branch to break $P$ into $P_1, \ldots, P_k$ such that $P \cap Z^n = P_1 \cup \ldots \cup P_k$.

In practice, $P$ is broken into $P \cap \{x : ax \geq b\}$ and $P \cap \{x : ax \leq b-1\}$ for some class of halfspaces.
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

**Branch-and-Cut** on input $P$,

Branch: break $P$ into $P_1, \ldots, P_k$

s.t. $\mathbb{PNZ}^n \subseteq P_1 \cup \ldots \cup P_k$

Cut: Refine $P_1, \ldots, P_k$ with additional cutting planes.
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

**Branch-and-Cut** on input $P$,

- **Branch**: break $P$ into $P_1, \ldots, P_k$
- **s.t.**: $PNZ^n \subseteq P_i \cup \ldots \cup P_k$

- **Cut**: Refine $P_1, \ldots, P_k$ with additional cutting planes
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

Branch-and-Cut on input $P$,

- **Branch**: break $P$ into $P_1, \ldots, P_k$
- **s.t.** $P_{NZ}^n \leq P_1 \cup \ldots \cup P_k$
- **Cut**: Refine $P_1, \ldots, P_k$ with additional cutting planes
Integer Programming

Modern IP Algorithms combine cutting planes with a branch-and-bound procedure.

**Branch-and-Cut** on input $P$:
- **Branch**: break $P$ into $P_1, ..., P_k$
  - s.t. $P_{NZ} \subseteq P_1 \cup ... \cup P_k$
- **Cut**: Refine $P_1, ..., P_k$ with additional cutting planes

Repeat
Stabbing Planes [BF1+18]

- Formalizes practical branch-and-cut as a proof system
Stubbing Planes \[BF1+18]\n
- Formalizes practical branch-and-cut as a proof system
- Extends DPLL to reason about linear inequalities
DPLL Refutation
\{x_1 \lor x_2, \overline{x}_1 \lor x_2, x_1 \lor \overline{x}_2, \overline{x}_1 \lor \overline{x}_2\}
DPLL Refutation

\{ x_1 \lor x_2, \bar{x}_1 \lor x_2, x_1 \lor \bar{x}_2, \bar{x}_1 \lor \bar{x}_2 \}
DPLL Refutation

\{ x_1 \lor x_2, \overline{x}_1 \lor x_2, x_1 \lor \overline{x}_2, \overline{x}_1 \lor \overline{x}_2 \}
DPLL Refutation

\{ x_1 \lor x_2, \overline{x}_1 \lor x_2, x_1 \lor \overline{x}_2, \overline{x}_1 \lor \overline{x}_2 \}
DPLL Refutation

\{ x_1 \lor x_2, \overline{x}_1 \lor x_2, x_1 \lor \overline{x}_2, \overline{x}_1 \lor \overline{x}_2 \}
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \left\{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \right\} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, \quad x_1 - x_2 \geq 0, \quad x_2 - x_1 \geq 0, \quad -x_1 - x_2 \geq -1, \quad 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, \ x_1 - x_2 \geq 0, \ x_2 - x_1 \geq 0, \ -x_1 - x_2 \geq -1, \ 0 \leq x_i \leq 1 \} \]
DPLL as Polytopes

\[ P = \{ x_1 + x_2 \geq 1, x_1 - x_2 \geq 0, x_2 - x_1 \geq 0, -x_1 - x_2 \geq -1, 0 \leq x_i \leq 1 \} \]

Refutation! \( \emptyset \) Derived at every Leaf of the tree

\[ P \]

\[ P \{ x_1 \leq 0 \} \]

\[ P \{ x_1 \geq 1 \} \]

\[ P \{ x_2 \leq 0 \} \]

\[ P \{ x_2 \geq 1 \} \]

\[ \emptyset \]

Empty Polytope
Stabbing Planes

\[ P = \{ Ax \geq b \} \quad \text{s.t. } \text{PNZ}^n = \emptyset \]
Stabbing Planes

\[ P = \{ Ax \geq b \} \text{ s.t. } P \cap \mathbb{Z}^n = \emptyset. \]
Stabbing Planes

\[ P = \{ Ax \geq b \} \text{ s.t. } P \cap \mathbb{R}^n = \emptyset. \]

\[ ax \leq b-1 \quad ax \geq b \]
Stabbing Planes

\[ P = \{ Ax \geq b \} \text{ s.t. } P \cap \mathbb{Z}^n = \emptyset. \]

\[ P \cap \{ ax \leq b - 1 \} \]
Stabbing Planes

\[ P = \{ Ax \gg b \} \text{ s.t. } P \cap \mathbb{R}^n = \emptyset . \]
Stabbing Planes

\[ P = \{ Ax \gg b \} \quad \text{s.t.} \quad \text{PnZ}^n = \emptyset. \]
Stabbing Planes

\[ P = \{ A\mathbf{x} \geq b \} \quad \text{s.t.} \quad \text{P has } \mathbb{Z}^n = \emptyset. \]

\[ a_1 \mathbf{x} \leq b_1 - 1 \]
\[ a_2 \mathbf{x} \leq b_2 - 1 \]
\[ a_1 \mathbf{x} \geq b_1 \]
\[ a_2 \mathbf{x} \geq b_2 \]

[Diagram of a geometric figure]
Stabbing Planes

\[ P = \{Ax \geq b\} \text{ s.t. } P \cap \mathbb{Z}^n = \emptyset. \]
Stabbing Planes

\[ P = \{ Ax \geq b \} \quad \text{s.t. } P \cap Z^n = \emptyset. \]

\[ P \cap \{ ax \leq b \} \quad \text{Refutation} \]

\[ \emptyset \]

Empty polytope \( \emptyset \) deduced at every leaf proves \( P \cap Z^n = \emptyset \).
Stabbing Planes

\[ P = \{ Ax \geq b \} \text{ s.t. } \text{PNZ}^n = \emptyset. \]

**Refutation**

Empty polytope \( \emptyset \) deduced at every leaf proves \( \text{PNZ}^n = \emptyset \)

**Size:** \# of queries
Stabbing Planes

\[ P = \{ Ax \geq b \} \text{ s.t. } \mathbb{P}^{n \mathbb{P}^{Z_n}} = \emptyset. \]

Refutation

Empty polytope \( \emptyset \) deduced at every leaf proves \( \mathbb{P}^{n \mathbb{P}^{Z_n}} = \emptyset \)

Size: \# of queries \( \times \) bit-size [DT20]
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
Stubbng Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Claim: path like SP = CP
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax \leq b-1, ax > b)\) at \(P\) is pathlike if \(\text{PN}(ax \leq b-1) = \emptyset\) or \(\text{PN}(ax > b)\)

Claim: Pathlike SP = CP
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax < b - 1, ax \geq b)\) at \(P\) is pathlike if \(\Pi(ax < b - 1) = \emptyset\) or \(\Pi(ax \geq b)\)

Claim: pathlike SP = CP
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query $(ax \leq b-1, ax \geq b)$ at $P$ is pathlike
  if $\text{PN}\{ax \leq b-1\} = \emptyset$ or $\text{PN}\{ax \geq b\}$

Claim: pathlike SP = CP
**Stubbing Planes vs Cutting Planes**

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

**Pathlike SP:** A SP query \((ax \leq b, ax > b)\) at \(P\) is pathlike if \(P \cap \{ax \leq b\} = \emptyset\) or \(P \cap \{ax > b\} = \emptyset\)

**Cllm:** Pathlike SP = CP

Pathlike SP Proof:

- \(\emptyset\)
- \(\emptyset\)
- \(\emptyset\)
- \(\emptyset\)

Proof:

\[\emptyset \rightarrow \emptyset \rightarrow \emptyset \rightarrow \emptyset\]
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax \leq b - 1, ax \geq b)\) at \(P\) is pathlike if \(\text{PN}\{ax \leq b - 1\} = \emptyset\) or \(\text{PN}\{ax \geq b\}\)

Clm: path like SP = CP

Pf: Show each CG-cut is a pathlike query
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax \leq b-1, ax \geq b)\) at \(P\) is pathlike if \(\text{PN}\{ax \leq b-1\} = \emptyset\) or \(\text{PN}\{ax \geq b\}\)

Claim: Pathlike SP = CP

Proof: Show each CG-cut is a pathlike query
\(ax \geq b\) is a CG-cut for \(P\) if \(ax \geq b\) is valid for \(P\)
Stabbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax \leq b - 1, ax \geq b)\) at \(P\) is pathlike if \(\text{PN}(ax \leq b - 1) = \emptyset\) or \(\text{PN}(ax \geq b)\)

Claim: Pathlike SP = CP

Proof: Show each CG-cut is a pathlike query
  \(ax \geq \lceil b \rceil\) is a CG-cut for \(P\) if \(ax \geq b\) is valid for \(P\)

\[\Rightarrow \text{PN}(ax \geq \lceil b \rceil - 1) = \emptyset\]
Stubbing Planes vs Cutting Planes

- Branch-and-cut allows cutting planes deductions
  - Superfluous as SP can simulate CP

Pathlike SP: A SP query \((ax \leq b-1, ax \geq b)\) at \(P\) is pathlike if \(PN\{ax \leq b-1\} = \emptyset\) or \(PN\{ax \geq b\}\)

Claim: pathlike SP = CP

Proof: Show each CG-cut is a pathlike query
- \(ax \geq b-1\) is a CG-cut for \(P\) if \(ax \geq b\) is valid for \(P\)
  - \(PN\{ax \geq b-1\} = \emptyset\) & \((ax \geq b-1, ax \geq b)\) is pathlike
Stubbing Planes vs Cutting Planes

Q: Can we separate CP from SP?
Stubbing Planes vs Cutting Planes

Q Can we separate CP from SP?

Candidate: Tseitin formulas
Stabbing Planes vs Cutting Planes

Q Can we separate CP from SP?

 Candidate: Tseitin formulas
 -> quasipoly-size SP proofs of Tseitin [BFIt18]
Stubbing Planes vs Cutting Planes

Q. Can we separate CP from SP?

- Candidate: Tseitin formulas

  - Quasipoly-size SP proofs of Tseitin \([\text{BFIT}18]\)

  - Conjectured to be hard for CP
Stubbing Planes vs Cutting Planes

Q: Can we separate CP from SP?

- Candidate: Tseitin formulas
  - Quasipoly-size SP proofs of Tseitin [BFI+18]
  - Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin
Stabbing Planes vs Cutting Planes

Q. Can we separate CP from SP?

- Candidate: Tseitin formulas
  - Quasipoly-size SP proofs of Tseitin [BFI+18]
  - Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin

- Translate the SP proof of Tseitin into CP
Stubbing Planes vs Cutting Planes

Q Can we separate CP from SP?
  ▶ Candidate: Tseitin formulas
    ➔ quasi-poly-size SP proofs of Tseitin [BFI+18]
    ➔ Conjectured to be hard for CP

[DT20] There are quasi-poly size CP proofs of Tseitin
  ▶ Translate the SP proof of Tseitin into CP

Q Can every SP proof be translated into CP?
Claim: Every $SP^*$ proof can be quasipolynomially translated into $CP$.
Stubbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

coefficients in queries are quasi-poly bounded
Stubbing Planes vs Cutting Planes

Thm: Every $SP^*$ proof can be quasipolynomially translated into $CP$

Pf: (1) $CP =$ pathlike $SP =$ facelike $SP$
Stubbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: (1) CP = pathlike SP = facelike SP

Pathlike SP: A query $\langle ax < b-1, ax \geq b \rangle$ is pathlike if $PN\{ax < b-1\} = \emptyset$ or $PN\{ax \geq b\} = \emptyset$
Stabbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: (1) CP = pathlike SP = facelike SP

Pathlike SP: A query \((ax \leq b-1, ax \geq b)\) is pathlike if
\[ \operatorname{PN}\{ax < b-1\} = \emptyset \text{ or } \operatorname{PN}\{ax \geq b\} = \emptyset \]

Facelike SP: A query \((ax \leq b-1, ax \geq b)\) is facelike if
\[ \operatorname{PN}\{ax < b-1\} = \emptyset \text{ or } \operatorname{PN}\{ax \geq b\} = \emptyset \]
Stubbing Planes vs Cutting Planes

Thm: Every SP* proof can be quasipolynomially translated into CP

Pf: (1) CP = pathlike SP = facelike SP

Pathlike SP: A query \((ax \leq b, ax > b)\) is pathlike if
\[ PN\{ax \leq b\} = \emptyset \text{ or } PN\{ax > b\} = \emptyset \]

Facelike SP: A query \((ax \leq b, ax > b)\) is facelike if
\[ PN\{ax < b\} = \emptyset \text{ or } PN\{ax > b\} = \emptyset \]
\[
\text{ie } PN\{ax \leq b\} \text{ or } PN\{ax > b\} \text{ is a face.} \]
Stubbing Planes vs Cutting Planes

Thm: Every $SP^*$ proof can be quasipolynomially translated into CP

Pf: a) CP = pathlike $SP$ = facelike $SP$

b) Any $SP^*$ proof can be made facelike with a quasi-poly blowup in size.
Stubbing Planes vs Cutting Planes

Thm: Every $SP^*$ proof can be quasipolynomially translated into $CP$

Pf: (a) $CP =$ pathlike $SP =$ facelike $SP$
(b) Any $SP^*$ proof can be made facelike with a quasi-poly blowup in size.
Stubbing Planes vs Cutting Planes

Thm: Every $SP^*$ proof can be quasipolynomially translated into $CP$

Pf: (a) $CP = path-like SP = facelike SP$

(b) Any $SP^*$ proof can be made facelike with a quasi-poly blowup in size.
Stubbing Planes vs Cutting Planes

Thm: Every $SP^*$ proof can be quasipolynomially translated into $CP$

Cor: Exponential lower bounds on $SP^*$
Stubbing Planes vs Cutting Planes

Thm: Every \( SP^* \) proof can be quasipolynomially translated into \( CP \)

Cor: Exponential lower bounds on \( SP^* \)

Cor: Cutting Planes has quasi-poly size proofs of any system of linear equations over a finite field
Depth of CP Proofs of Tseitin

Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep
Depth of CP Proofs of Tseitin

- Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep.
- Any CNF Can be refuted in depth n.
Depth of CP Proofs of Tseitin

- Both (ours and [DT20]) CP proofs of Tseitin are quasi-polynomially deep.
- Any CNF can be refuted in depth n.

Conjecture: There is a family of CNFs \( \{F_n\} \) which has quasi-poly CP proofs, but any quasi-poly CP proof requires superlinear depth.
Towards the Conjecture

Conjecture: There is a family of CNFs \( \{F_n\} \) which has quasi-poly CP proofs, but any quasi-poly CP proof requires superlinear depth.
Towards the Conjecture

Conjecture: There is a family of CNFs \( \{F_n\} \) which has quasi-poly CP proofs, but any quasi-poly CP proof requires superlinear depth.

\( \square \) Requires a better understanding of CP depth.
Towards the Conjecture

Conjecture: There is a family of CNFs \( \{F_n\} \) which has quasi-poly CP proofs, but any quasi-poly CP proof requires superlinear depth.

- Requires a better understanding of CP depth
- A new geometric technique for CP depth bounds
Towards the Conjecture

Conjecture: There is a family of CNFs \( \{F_n\} \) which has quasi-poly CP proofs, but any quasi-poly CP proof requires superlinear depth.

- Requires a better understanding of CP depth
- A new geometric technique for CP depth bounds

Thm: Tseitin requires \( \Omega(n) \) depth to refute in semantic CP
Open Problems

Can CP \( p \)-simulate \( SP^* \)?
Open Problems

- Can CP \( p \)-simulate SP*?
- Can CP (quasipolynomially) simulate SP?
Open Problems

- Can CP \( \text{p-simulate} \) \( SP^* \)?
- Can \( CP \) (quasipolynomially) simulate \( SP \)?
- Can \( CP^* \) (quasipolynomially) simulate \( SP^* \)?
Open Problems

- Can CP $p$-simulate $SP^*$?
- Can CP (quasipolynomially) simulate $SP$?
- Can $CP^*$ (quasipolynomially) simulate $SP^*$?
- Can $SP$ or $CP$ simulate dag-like $SP$ ($R(CP)$)?
Open Problems

- Can CP \( p \)-simulate \( SP^* \)?
- Can CP (quasipolynomially) simulate \( SP \)?
- Can \( CP^* \) (quasipolynomially) simulate \( SP^* \)?
- Can \( SP \) or \( CP \) simulate dag-like \( SP \) (\( R(CP) \))?
  \[ \rightarrow \text{CP cannot \( p \)-simulate \( R(CP) \)} \ [\text{ABEOZ}] \]
Open Problems

- Can CP \( p \)-simulate \( SP^* \)?
- Can CP (quasipolynomially) simulate \( SP \)?
- Can \( CP^* \) (quasipolynomially) simulate \( SP^* \)?
- Can \( SP \) or CP simulate dag-like \( SP \) (\( R(CP) \))? 
  \[ \rightarrow \] CP cannot \( p \)-simulate \( R(CP) \) [ABEOZ]
- Resolve the Conjecture